

What is claimed is:

1. A method for dynamically loading program logic, comprising:

(a) launching an application from client computer that issues a request;

(b) receiving the request at the server and transferring a configuration file to the client computer based on said request, wherein the configuration file comprises a program logic file name, and a program logic file address, and the program logic file address corresponds to a storage apparatus where the program logic file corresponding to the program logic file name is located, said program logic file comprises the program logic required to execute the application;

(c) receiving the configuration file at client computer;

(d) linking to the storage apparatus corresponding to the program logic file address and downloading the program logic file which corresponds to the program logic file name, according to the program logic file address corresponding to the program logic file name in the configuration file from client computer; and

(e) executing the application in accordance with program logic in the program logic file at the client computer.

2. The method according to claim 1, wherein the storage apparatus is an external server.

3. The method according to claim 1, wherein the server further comprises a storage apparatus.

4. The method according to claim 1, wherein the client computer further comprises a client computer storage apparatus used

3 to store the configuration file.

1 5. The method according to claim 1, wherein the client computer
2 further comprises a client computer storage apparatus used
3 to store the program logic file.

1 6. The method according to claim 1, wherein the client computer
2 further comprises a program interpreter and executes default
3 program logic in the default program logic file while
4 launching the application from the client computer, and in
5 step (e), the program interpreter executes the following
6 steps:

7 parsing program logic of the program logic file;

8 terminating default program logic in the default program
9 logic file; and

10 executing program logic in the program logic file, thereby
11 completing application execution.

12 7. The method according to claim 6, wherein the client computer
13 receives a program logic file replacing the default program
14 logic file.

1 8. The method according to claim 1, wherein the client computer
2 further comprises a program interpreter, which executes the
3 following steps in step (e):

4 parsing program logic of the program logic file; and

5 executing program logic in the program logic file, thereby
6 completing application execution.

1 9. The method according to claim 1, wherein the configuration
2 file further comprises a timer used for initializing the
3 execution of steps (a) and (e) from client computer at preset

4 times.

1 10. The method according to claim 1, wherein the configuration
2 file further comprises a timer used for initializing the
3 execution of steps (a), (d) and (e) from client computer
4 at preset times.

1 11. The method according to claim 1, wherein the client computer
2 further comprises a driver module used to initialize the
3 execution of steps (a) and (e) from client computer.

1 12. A method for dynamically loading program logic comprising:

2 (a) launching an application and making a request from
3 the client computer, wherein upon launch, a first program
4 logic is executed and the client computer comprises a first
5 program logic file comprising the first program logic and
6 a first configuration file, comprising first version
7 recognition code corresponding to the first program logic
8 file;

9 (b) receiving the request at server and transferring a
10 second configuration file to the client computer based on
11 said request, and wherein the second configuration file
12 comprises a program logic file name, a program logic file
13 address and a second version recognition code, wherein the
14 program logic file address corresponds to a storage apparatus
15 where the second program logic file corresponding to the
16 file name of program logic is located, and the second program
17 logic file comprises a second program logic required for
18 application execution, wherein the second version
19 recognition code corresponds to the second program logic;

20 (c) receiving the second configuration file from client
21 computer;

(d) determining whether the second version recognition code and the first version recognition code are identical at the client computer, if yes, the application execution proceeds in accordance with the first program logic, if not, executing step (e);

(e) linking to the storage apparatus corresponding to the program logic file address and downloading the second program logic file corresponding to program logic file name in the second configuration file from the client computer; and

(f) executing the application at the client computer according to the second program logic in the second program logic file.

13. The method according to claim 12, wherein the storage apparatus is an external server.

14. The method according to claim 12, wherein the server further comprises a storage apparatus.

15. The method according to claim 12, wherein the client computer replaces the first configuration file with the second configuration file.

16. The method according to claim 12, wherein the client computer further comprises a program interpreter which executes the following steps in step (f):

parsing second program logic in the second program logic file;

terminating first program logic; and

executing second program logic in the second program logic

file and completing application execution.

17. The method according to claim 12, wherein the client computer replaces the first program logic file with the second program logic file.

18. The method according to claim 12, wherein the configuration file further comprises a timer used to initialize the execution of steps (a) and (f) at the client computer at preset times.

19. The method according to claim 12, wherein the configuration file further comprises a timer used to initialize the execution of steps (a), (e) and (f) at the client computer at preset times.

20. The method according to claim 12, wherein the client computer further comprises a driver module used to initialize the execution of steps (a) and (f) at the client computer.

21. A system for dynamically loading program logic comprising:

a server, comprising:

a plurality of configuration files, each configuration file comprising a program logic file name and a program logic file address, wherein the program logic file address corresponds to a storage apparatus, where the program logic file corresponding to the program logic file name is located, and the program logic file comprises the program logic required for application execution;

a client computer, comprising:

a client computer storage apparatus used to store the

application;

a launch module used to launch the application;

a communication module used to receive the
configuration files transferred from the server;

a download module used to link to the storage apparatus
corresponding to the program logic file address and
downloads the program logic file corresponding to the
program logic file name in the configuration file from
client computer; and

an executing module used for application execution
according to program logic in the program logic file.

22. The system according to claim 21, wherein the storage
apparatus is an external server.

23. The system according to claim 21, wherein the server further
comprises a storage apparatus.

24. The system according to claim 21, wherein the client computer
stores received configuration files in the client computer
storage apparatus.

25. The system according to claim 21, wherein the client computer
stores program logic files in the client computer storage
apparatus.

26. The system according to claim 21, wherein the client computer
further comprises a program interpreter stored in the client
computer storage apparatus and a default program logic file
stored in the client computer storage apparatus, and the
default program logic file comprises default program logic,
which is launched when the application is launched by a launch

module at the client computer, and the program interpreter is used to parse the program logic in the received program logic file, terminate default program logic, execute program logic in the program logic file and complete the application execution.

27. The system according to claim 26, wherein the client computer replaces the default program logic file with a received program logic file.

28. The system according to claim 21, wherein the client computer further comprises a program interpreter stored in the client computer storage apparatus used to parse program logic in the program logic file, executing the program logic in the program logic file thereby completing application execution.

29. The system according to claim 21, wherein the configuration file further comprises a timer used to initialize the launch module, communication module, download module and executing module at the client computer at preset times.

30. The system according to claim 21, wherein the configuration file further comprises a timer used to initialize the launch module, download module and executing module at the client computer at preset times.

31. The system according to claim 21, wherein the client computer further comprises a driver module, used to initialize launch module, communication module, download module and executing module at the client computer.

32. A system for dynamically loading program logic comprising:
a server comprising:

3 a plurality of second configuration files, wherein each
4 second configuration file comprises a program logic file
5 name, a program logic file address, and a second version
6 recognition code, and the program logic file address
7 corresponds to a storage apparatus where the second program
8 logic file corresponding to the program logic file name is
9 located, and the second program logic file comprises program
10 logic required for application execution, and the second
11 version recognition code corresponds to the second program
12 logic file;

13 a client computer comprising:

14 a client computer storage apparatus used to store the
15 application, a first program logic file and a first
16 configuration file, wherein the first program logic file
17 comprise first program logic, the first configuration
18 file comprises a first version recognition code,
19 corresponding to the first program logic file;

20 a launch module used to launch the application and
21 simultaneously execute first program logic when the
22 application is launched;

23 a communication module used to receive the second
24 configuration file from the server;

25 a download module used to link to the storage apparatus
26 corresponding to the program logic file address and
27 downloads the second program logic file corresponding
28 to the program logic file name in the second configuration
29 file at client computer;

30 an executing module used to execute an application
31 in accordance with first or second program logic; and

a determining module used to determine whether the second version recognition code and the first version recognition code are identical, if yes, the execution module executes the application in accordance with the first program logic, if not, the download module downloads the second program logic file corresponding to the program logic file name, and the execution module is initialized to execute the application in accordance with the second program logic in the second program logic file.

33. The system according to claim 32, wherein the storage apparatus is an external server.

34. The system according to claim 32, wherein the server further comprises a storage apparatus.

35. The system according to claim 32, wherein if the first version recognition code differs from the second version recognition code, the client computer replaces the first program logic file with the downloaded second program logic file.

36. The system according to claim 32, wherein if first version recognition code differs from second version recognition code, the client computer replaces the first configuration file with the second configuration file.

37. The system according to claim 32, wherein the client computer further comprises a program interpreter stored in the client computer storage apparatus which is used to parse the second program logic in the second program logic file, to terminate the first program logic, and to execute the second program logic in the second program logic file for completing the application execution.

38. The system according to claim 32, wherein the second

configuration file further comprises a timer used to initialize the launch module, the communication module, the download module, the determining module and the executing module at the client computer at preset times.

39. The system according to claim 32, wherein the second configuration file further comprises a timer used to initialize the launch module, the download module and the executing module at the client computer at preset times.

40. The system according to claim 32, wherein the client computer further comprises a driver module used to initialize the launch module, the communication module, the download module, the determining module and the executing module at the client computer.